
Django-thumbnail-maker Documentation

Release dev

Tamara Khalbashkeyeva

Jul 01, 2017

Contents

1	Django-thumbnail-maker application	3
1.1	Features	3
1.2	How to Use	3
1.2.1	Get the code	3
1.2.2	Configure your project	3
1.2.3	Set up your model's field	4
1.2.4	Templates usage	4
1.2.5	Management commands usage	5
2	Indices and tables	7

Contents:

Django-thumbnail-maker application

Auto-generator of thumbnails for Django, using sorl-thumbnail.

Features

- Auto-generates thumbnails using sorl-thumbnail while uploading (saving) images
- You can use any engines & plugins you usually use with sorl-thumbnail
- The application does not replace thumbnail templatetag and you can use everything you want from sorl-thumbnail
- Command to auto-generate missing thumbnails

How to Use

Get the code

Get the code for the latest stable release use pip:

```
$ pip install django-thumbnail-maker
```

Configure your project

Register 'thumbnail_maker', in the `INSTALLED_APPS` section of your project's settings anywhere after 'sorl.thumbnail' app (assume you already installed sorl-thumbnail):

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.admin',
```

```
'django.contrib.sites',
'django.contrib.comments',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.contenttypes',
...
'sorl.thumbnail',
...
'thumbnail_maker',
)
```

Set up required thumb formats in `THUMBNAI_MAKER_FORMATS` dictionary. Keys are the names for your formats (could be any string). Values are tuples of length 2: first element is a geometry string (used in `sorl-thumbnail`), second is a dictionary with options (crop, quality, padding, format, etc.):

```
THUMBNAI_MAKER_FORMATS = {
    'banner': ('400x300', {'crop': 'center',
                          'quality': 90}),
    '50x50': ('50x50', {}),
    'any_name': ('5x277', {'padding': True})
}
```

You can set up `THUMBNAI_MAKER_DEBUG` setting. By default it is set to `False` in order to pass exceptions while saving models or while generating batch of thumbnails. This setting is not required. Usage:

```
THUMBNAI_MAKER_DEBUG = True
```

Set up your model's field

Use `ImageWithThumbnailsField` and `thumbs` option, where `thumbs` is a tuple of thumb format names (keys from `THUMBNAI_MAKER_FORMATS` dictionary):

```
from django.db import models
from thumbnail_maker.fields import ImageWithThumbnailsField

class Item(models.Model):
    image = ImageWithThumbnailsField(
        upload_to='somewhere',
        thumbs=('banner', '50x50'),
    )
```

Templates usage

All of the examples assume that you first load the `thumbnail_maker` template tag in your template:

```
{% load thumbnail_maker %}
```

A simple usage:

```
{% usethumbnail item.image "banner" as im %}

{% endusethumbnail %}

{% usethumbnail item.image "50x50" as im %}
```



```

{% endusethumbnail %}
```

You can also use string paths instead of image objects:

```
{% usethumbnail "dummy/image.png" "50x50" as im %}
  
{% endusethumbnail %}
```

Management commands usage

Django-thumbnail-maker comes with a `manage.py` command to generate missing thumbs. You can use it while

```
./manage.py make_thumbnails <app>.<model> <field>
```

In case you want to make all thumbs replacing old ones, use `--force` option:

```
./manage.py make_thumbnails --force <app>.<model> <field>
```


CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`